

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **11195013 A**

(43) Date of publication of application: 21 . 07 . 99

(51) Int. Cl.

G06F 17/21

(21) Application number: **10012116**

(71) Applicant: **FUJI XEROX CO LTD**

(22) Date of filing: **06 . 01 . 98**

(72) Inventor: **ITO SATOSHI**

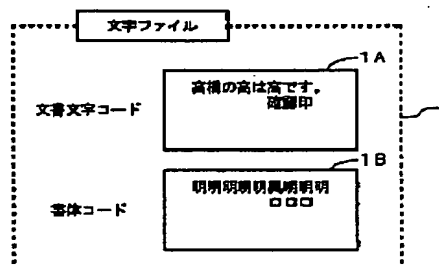
(54) DOCUMENT PROCESSOR

(57) Abstract:

PROBLEM TO BE SOLVED: To provide a document processor which can deal with a document containing a character and a pattern, which are expressed by an irregular code.

SOLUTION: A document file 1 being a processing object is constituted of a character code and a handwriting code. The handwriting code corresponds to the character code by one to one. A font name is specified from a font information table with the handwriting code as a key and font data are generated based on the font name and the character code. When the font name corresponding to the handwriting code does not exist in the table, the table outputs the code showing the alternate font name. Unique character can be outputted by the irregular code by making data of an external character or the pattern correspond to the alternate code so as to register it.

COPYRIGHT: (C)1999,JPO



Japanese Patent Application Laid-Open No. 11-195013

[ABSTRACT]

[OBJECT] The present invention is to provide a word processor which can handle a document including a character represented with a code, a pattern or the like rather than a code standard.

[SOLVING MEANS] A document file 1 to be processed comprises a character code and a font code. The font code corresponds to the character code one to one. A font name is specified from a font information table using the font code as a key, and font data is generated on the basis of the font name and the character code. When the font name corresponding to the font code does not exist in the table, the table outputs a code showing a substitute font name. By registering data corresponding to a pattern or an external character in the substitute code in advance, a unique character or the like can be output by a code rather than a code standard.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-195013

(43) 公開日 平成11年(1999) 7月21日

(51) Int.Cl.*

識別記号

F I

G 0 6 F 17/21

G 0 6 F 15/20

5 6 2 P

審査請求 未請求 請求項の数 6 F D (全 10 頁)

(21) 出願番号 特願平10-12116

(22) 出願日 平成10年(1998) 1月6日

(71) 出願人 000005496

富士ゼロックス株式会社

東京都港区赤坂二丁目17番22号

(72) 発明者 伊藤 聡

埼玉県岩槻市府内3丁目7番1号 富士ゼ

ロックス株式会社内

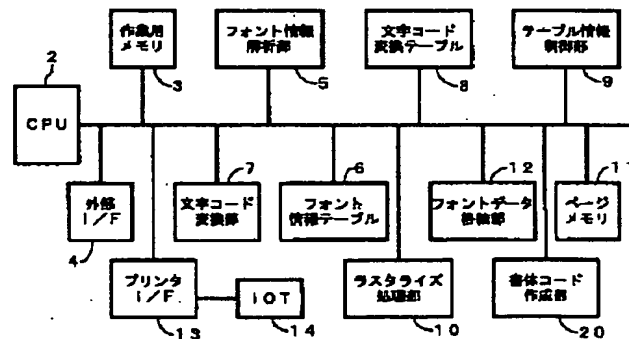
(74) 代理人 弁理士 田中 香樹 (外1名)

(54) 【発明の名称】 文書処理装置

(57) 【要約】

【課題】 コード規格外のコードで表現された文字やパターン等を含む文書を扱うことができる文書処理装置を提供する。

【解決手段】 処理対象となる文書ファイル1は文字コードと書体コードとからなる。書体コードは文字コードに一对一に対応している。書体コードをキーとしてフォント情報テーブルからフォント名を特定し、そのフォント名と文字コードとに基づいてフォントデータを発生させる。書体コードに対応するフォント名が前記テーブルに存在しない場合、該テーブルは代替のフォント名を示すコードを出力する。この代替のコードに予め外字またはパターンのデータを対応づけて登録しておくことによって、コード規格外のコードによってユニークな文字等を主力させることができる。



【特許請求の範囲】

【請求項1】 文書ファイル中の文字コード列および該文字コード列の個々の文字コードに対応づけられた書体コードからなる書体コード列を解釈して前記文字コードで代表されるフォントデータを出力する文書処理装置において、

前記書体コードを検索キーとしてフォント名を出力するフォント情報テーブルと、

前記文字コードおよび前記フォント名を検索キーとしてフォントデータを出力するフォントデータ格納手段とを具備したことを特徴とする文書処理装置。

【請求項2】 前記フォント情報テーブルが、前記フォントデータ格納手段に設定されていないフォント名に代わるデフォルトのフォント名を保持していて、該設定されていないフォント名を示す書体コードが入力されたときには前記デフォルトのフォント名を出力するように構成されたことを特徴とする請求項1記載の文書処理装置。

【請求項3】 前記フォント情報テーブルが、前記フォントデータ格納手段に設定されていないフォント名を示す書体コードが入力されたときに前記文字コードまたは書体コードを置換文字コードまたは置換書体コードに変換するための判断を行わせる指示情報を出力するように構成されているとともに、前記指示情報に応答して、複数の文字コードからなる文字コード列が、予定の変換条件と一致しているか否かを判断し、一致していたときには前記文字コード列を予定の置換文字コードに変換し、一致していないときには前記書体コードを予定の置換書体コードに変換する文字コード変換手段を具備したことを特徴とする請求項1または2記載の文書処理装置。

【請求項4】 前記文字コード変換手段が、書体コードおよび該書体コードに対応づけて設定された文字コード列ならびに置換文字コードおよび置換書体コードが記述された文字コード変換テーブルを具備し、前記文字コード列の判断ならびに置換文字コードおよび置換書体コードへの変換を、前記文字コード変換テーブルを使用して実行するように構成されたことを特徴とする請求項3記載の文書処理装置。

【請求項5】 前記フォント情報テーブルおよび文字コード変換テーブルを制御するための対象テーブル名、制御内容、被制御データを含むテーブル制御データを受信する手段と、前記テーブル制御データに記述された制御内容および被制御データに基づき、対象となるテーブルのデータを出力・更新・追加・削除するテーブル情報制御手段とを具備したことを特徴とする請求項4記載の文書処理装置。

【請求項6】 文書ファイル上の文字列を表示する表示手段と、

表示された文字列を構成する文字、および該文字の書体

を指定するための入力手段と、

前記指定された文字以外の文字にはデフォルトの書体コードを設定し、前記指定された文字には指定された書体の書体コードを設定する書体コード生成手段と、

前記文字の文字コードおよび前記書体コード生成手段で設定された書体コードを登録する記憶手段とを具備したことを特徴とする請求項1～5のいずれかに記載の文書処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、コードで表現される文字やパターン等を含む文書进行处理する文書処理装置に関し、特に、コード規格外のコードで表現された文字やパターン等を含む文書を扱うことができる文書処理装置に関する。

【0002】

【従来の技術】ワードプロセッサやコンピュータ等の文書処理装置では、そこで処理される文書を構成する文字や、その文字に関する種々の修飾情報はすべてコード化されている。これら文字や修飾情報を表現するコードは、異機種間でのデータの互換性をもたせるために規格化されている。コード規格は数多く存在するが、同一規格内のコードであれば、文書処理装置の機種に関わりなく文書の互換性は維持される。コードの規格としては、JISコード体系がよく知られている。

【0003】しかし、前記コード規格の多くは、常用漢字等、頻繁に使用される文字をコード化したものであり、人名漢字や旧漢字等が規格から外されていることが多い。規格から外れた文字やパターン等の外字については、例えばJISコード体系の未使用のコード領域に割り当てる等、製造者が定義することができる。

【0004】図14は、特開平6-259426号公報に記載された文字処理装置の要部を示すブロック図である。同図において、処理文書中の文字データ100はフォント名と文字コードとで表現されている。この文字データ100が通常文字（規格コード内の文字）である場合はフォントファイル110から前記フォント名と文字コードとに従ってフォントデータを読み出し、出力装置に供給する。

【0005】一方、文字データ100が外字である場合は相対テーブル120を参照して外字ファイル130内の外字格納位置を引き出す。そして、この外字格納位置に基づいて外字ファイル130から外字フォントデータを読み出して出力装置に供給する。この例では、フォント名「ゴシック」と文字コード「2147」とから格納位置「1」が引き出され、外字「あ」が出力される。

【0006】図15は、特開平7-141332号公報に記載された文書作成装置の要部を示すブロック図である。この装置では、文書中の文字を、JISコード上ではこの文字と同じ種類の文字でありながら字体が異なる

10

20

30

40

50

もの、つまり異字体に置き換える。同図において、処理文書中の文書データ200は文字コードのみで表現されている。この文書データ200のうち4文字目の文字データ「3548」で表現されている文字を異字体に置き換える場合、表示画面上のこの文字が指示されると、文字／異字体テーブル210が参照されてポイントが引き出される。ポイントはデータ格納領域220上の格納位置を示す。そして、このポイントで示されたデータ格納領域220の格納位置に異字体を表すコードつまり変換後文字コードを格納し、文字コード「3548」を該ポイント「0002」に置き換える。文書データ230は置換後の文字コードを含む文字コード列からなる。

【0007】このように置き換えられた文字を出力する場合は、文字コードの上位3ビットが「000」であるか否かによってポイントであるか通常の文字コードであるかを判断し、ポイントであればそのポイントが示すデータ格納領域220の格納位置から変換後文字コードを読み出す。

【0008】図16は、特開平7-105195号公報に記載された字形管理装置の要部を示すブロック図である。この装置では、テーブルを参照して異字体を認識するのではなく、コード規格に適合した文字概念コードと、文字種類コードとで処理文書中の文字データを表現している。図示のように、2つの文字データ300、310は、共通の文字概念コード「3548」を有しているが、文字種類コードが「明朝体」と「ゴシック異字体」とである点で相違する。文字a、300a、および310aは、それぞれ文字概念コードのみで表現された文字ならびに文字データ300および310で表現された文字である。

【0009】

【発明が解決しようとする課題】上記従来の装置はいずれも次のような問題点があった。特開平6-259426号公報に記載された装置では、規格コード以外のコードを使用しているため、異機種で処理したときはコードが理解されないか、希望した文字とは全く異なる文字が出力されるという問題点がある。

【0010】また、特開平7-141332号公報に記載された装置では、多数のテーブルを使用しているため、処理が複雑になるうえ、この装置で作成した文書を、テーブルを有していない他の機種で出力した場合には、解釈されないという問題点がある。

【0011】さらに、特開平7-105195号公報に記載された装置で文字種類コードによって表現している異字体は、「偏」が同じで「旁」が異なる同じ読みまたは同じ意味の文字、例えば、同公報に記載されている魚の「サバ」を意味する二種類の漢字等である。該公報に記載された装置では、このような漢字以外の記号やロゴマークのようなイメージパターンを規格コードで表現することができない。したがって、これら記号やロゴマ

ク等に対応するためには規格コード以外のコードで表現するしかなく、結局、前二者と同様、異機種で処理した場合は、全く意味が通じない文書になって出力されるおそれがある。

【0012】本発明は、上記問題点を解消し、記号やロゴマーク等をコードによって表現できるようにするとともに、これらのコードを含む文書を異なる文書作成環境間で交換した場合に、少なくとも文書の意味が不明となることを防止することができる文書処理装置を提供することを目的とする。

【0013】

【課題を解決するための手段】上記の課題を解決し、目的を達成するための本発明は、文書ファイル中の文字コード列および該文字コード列の個々の文字コードに対応づけられた書体コードからなる書体コード列を解釈して前記文字コードで代表されるフォントデータを出力する文書処理装置において、前記書体コードを検索キーとしてフォント名を出力するフォント情報テーブルと、前記文字コードおよび前記フォント名を検索キーとしてフォントデータを出力するフォントデータ格納手段とを具備した点に特徴がある。

【0014】この特徴によれば、文字コードおよび書体コードで指定された書体のフォントデータを使用して出力装置から文書が出力される。文字コードはコード規格に合致したものを使用した場合であっても、書体コードによって、コード規格に含まれない文字やパターンを前記文字コードに対応させて出力させることができる。

【0015】

【発明の実施の形態】以下、本発明の実施形態を図面を参照して詳細に説明する。まず、本実施形態の文書処理装置で取り扱われるデータの構成を説明する。図2は入力データの構成を示す文書ファイルの一例である。文書ファイル1上のデータは文書文字コード（以下、単に「文字コード」）1Aと書体コード1Bとからなる。なお、図2では文書ファイル1上のデータは、理解の容易のため、コードに代えて出力形態の文字で示している。文字コード1Aは、文書中の文字をコードで記述したものであり、コード規格内のコードが使用されている。書体コード1Bは、文字コード1A中の各文字に対して1対1で対応するフォント情報である。書体コード1Bは文書作成環境および文書出力環境に依存する。書体コード1B中の「明」は明朝体、「異」は異字体、「ロ」はロゴマークを表す。

【0016】図2の文字コード1Aおよび書体コード1Bで表された文書をプリンタ装置やディスプレイ装置に出力した例を図3に示す。同図(a)は書体コード1Bに従って出力した例であり、「高」は異字体の「高」で出力され、「確認印」は「高橋」の印影で出力されている。一方、書体コード1Bを参照せずに出力した場合は、文字コード1Aがコード規格に従ってそのまま変換

されるので、図3(b)に示すような出力結果となる。

【0017】図4は、処理中に使用されるテーブルの変更処理に用いられるテーブル制御データの一例を示す図である。該テーブル制御データには、対象テーブルが文字コード変換テーブルかフォント情報テーブルかを示すデータと、出力・更新・追加・削除等、制御内容を示すデータと、具体的なテーブルの構造つまりデータ内容とが記述されている。このテーブル制御データに基づく処理は後述する。

【0018】続いて、図2に例示した前記文書ファイル1を処理する文書処理装置の機能を説明する。図1は、本発明の一実施形態に係る文書処理装置の構成を示すブロック図である。同図において、CPU2は文書処理装置全体の制御および演算処理を行う。作業用メモリ3は処理中のデータを一時的に格納する。外部インタフェース4はホストコンピュータ等、外部の文書作成装置で作成された文書ファイル(図2参照)を受け入れる。

【0019】フォント情報解析部5は入力された文書ファイルの書体コードとフォント情報テーブル6とを照合し、文書を出力する際に使用するフォントを特定する。文字コード変換部7は入力された文書ファイルの文字コードのコード列が予定の条件(後述する)に該当している場合に、該文字コードおよび書体コードを、文字コード変換テーブル8を照合して所定の文字コードまたは書体コードに置換する。テーブル情報制御部9は処理中で使用するテーブル情報の出力・更新・追加・削除の制御を行う。

【0020】ラスライズ処理部10は使用するフォントと文字コードとを用いてページメモリ11上にイメージデータを展開する。フォントデータ格納部12は出力用のフォントデータを格納している。プリンタインタフェース13は、イメージ出力端末(IOT)14にイメージデータを出力するためのインタフェースである。書体コード作成部20は、フォントにデフォルトを使用することを前提として、書体コードを設定していない文字コードのみの文書ファイルに関して、個々の文字を指定することによって書体コードを生成する。

【0021】図5は、前記フォント情報テーブル6の一例を示す図である。同図に示すように、フォント情報テーブル6には、フォント名、フォントデータ格納部12上のフォントの格納場所、デフォルト、コードチェックが記述されている。ここで、デフォルトは、書体コードによって指示されたデータが存在しない場合に使用するフォント名を示す番号、コードチェックは文字コードの変換を行うかどうかの指示を示すフラグである。このフラグが立っている場合は、後述の手順により複数の文字コードからなる文字コード列のチェックを行い、文字コード変換テーブル8を使ってこの文字コード列を所定の文字コードに置換する。

【0022】図6は、前記文字コード変換テーブル8の

一例を示す図である。同図に示すように、文字コード変換テーブル8には、使用条件となる書体コードおよび文字コード列、置換文字コード、置換書体コードが記述されている。ここで、置換文字コードは条件一致時に使用される文字コードであり、置換書体コードは条件不一致時に使用される書体コードである。

【0023】以上の構成において、外部インタフェース4を通じて入力されたデータは作業用メモリ3に一旦格納される。入力されるデータには上述の文書ファイルとテーブル制御データ(図2、図4)とがある。文書ファイルはフォント情報解析部5に転送され、テーブル制御データはテーブル情報制御部9に転送される。フォント情報解析部5は、書体コードとフォント情報テーブル6の内容に基づいて使用するフォントを特定する。書体コードに対応するフォントがフォント情報テーブル6上に存在しない場合は、該フォント情報テーブル6上の記述に基づいて別のフォントを選択する。例えば、書体コードで「明朝異体」が示されていた場合、図5のフォント情報テーブル6ではデフォルト値に「1」が記述されているので、番号「1」の明朝体のフォントが特定される。

【0024】さらに、書体コードで「ロゴマーク」が指定されていた場合、図5のフォント情報テーブル6ではコードチェックフラグがセットされているので、文字コード変換部7での処理に移行する。文字コード変換部7では、入力データつまり文字コードおよび書体コードが文字コード変換テーブル8に記述された条件に一致しているか否かによって文字コードを置換する。

【0025】例えば、フォント情報テーブル6を参照した結果、該当する書体コードのコードチェックフラグがセットされていた場合は文字コード変換テーブル8を参照して文字コード列をチェックする。その結果、文字コード列が文字コード変換テーブル8の文字コード列と一致した場合は置換文字コードが選択されるし、一致しない場合は置換書体コードが選択される。

【0026】図7はコードチェックフラグが立っていた場合の処理の概要を示す図である。同図において、文字コード列「88c9, 9301」および書体コード列「8001, 8001」からなる文書データXが存在し、文字コード変換テーブルTaが設定されている。この場合、文字コード列「88c9, 9301」は文字コード変換テーブルTaに含まれているので、文字コード列「88c9, 9301」は置換文字コード「2001」に置換される。文字コード列が文字コード変換テーブルTaの文字コード列と一致している場合は書体コード「8001」は維持される。つまり、文書を示すコードは文字コード「2001」および書体コード「8001」に変換される(文書データX1)。

【0027】一方、文字コード列が文字コード変換テーブルTaの文字コード列と一致していない場合は文字コ

10

20

30

40

50

ード列「88c9, 9301」が維持され、書体コード「8001, 8001」は置換書体コードとしてのデフォルト値「4001, 4001」に変換される（文書データX2）。

【0028】ラスライズ処理部10では、書体コード（置換書体コードを含む）によって選択されたフォントの中から文字コードを検索キーとしてフォントデータ格納部12からフォントデータを取り出し、ページメモリ11上にイメージデータとして展開する。イメージデータはプリンタインタフェース13を通じてIOT14に送出される。

【0029】次に、上記処理をフローチャートを参照して説明する。図8は全体の処理を示すフローチャートである。ステップS1では作業用メモリ3上に存在しているデータのチェックを行う。文書コードと書体コードとが存在していた場合はステップS2に進み、書体コードに基づいてフォントの種類を特定するフォント情報解析を行う。ステップS3では種類が特定されたフォントの中から文字コードを検索キーとしてフォントデータを取得する。ステップS4では取得したフォントデータをページメモリ11に展開する。ステップS5ではIOT14からの要求に回答してページメモリ11からデータを出力する。

【0030】作業用メモリ3上に文字コードのみが存在していた場合にはステップS2を経由せず直接ステップS3に進む。この場合、ステップS3では、デフォルトのフォントの中からフォントデータを取得する。また、作業用メモリ3上にテーブル制御データが存在していた場合は、ステップS6に進み、テーブル制御データの指示に従ってテーブル制御処理を実行する。

【0031】続いて、前記フォント情報解析（ステップS2）、フォントデータ取得（ステップS3）、テーブル制御（ステップS6）の各処理をさらに詳細に説明する。まず、フォント情報解析を説明する。フォント情報解析では、各文字コード毎に、書体コードとフォント情報テーブルとを照合して書体つまりフォントの種類を特定していく。この照合作業において指定の書体コードにコードチェックフラグが対応付けられていれば文字コード毎の照合から文字コード列の照合に移行する。図9はフォント情報解析のフローチャートである。ステップS10において、書体コードをチェックする。ステップS11ではフォント情報テーブル6を参照して、指定の書体コードが、文字以外のマークつまり記号やロゴマークに対応しているか、つまりコードチェックフラグがセットされているか否かを判別する。マークが指定されていた場合はステップS12に進み、文字コード列をチェックし、その結果、文字コード列と文字コード変換テーブル8上の文字コード列とが一致しているかどうかを判別する（ステップS13）。すなわち、マークは複数の文字コードによって表され、各文字コードには同一の書体

コードを対応させている（図2の文書ファイル1では文書コード「確認印」にロゴマークを表す書体コード「ロロ」が対応している）ので、この複数の書体コードに対応する文字コード列が文字コード変換テーブル8上の文字コード列と一致しているかどうかを判断する。

【0032】文字コード列が一致していると判断されたならばステップS14に進み、文字コードとして書体コードに対応する置換文字コードを出力して、予め作成されて記憶されているマークのフォントデータを採用する。ステップS13で文字コード列が一致していないと判断されたならば、ステップS15に進み、置換書体コード、例えばデフォルトのフォントデータを選択する。

【0033】また、ステップS11でマーク指定がないと判断された場合は、ステップS16に進み、書体コードで指定されたフォントが存在するか否かをフォント情報テーブル6を参照して判断する。指定フォントがある場合は、ステップS17に進み、指定された書体コードに従って指定フォントを採用する。指定フォントがない場合は、ステップS18に進み、デフォルトフォントを選択する。

【0034】続いて、前記フォントデータ取得について詳述する。なお、この処理に入る時点ではすでにフォントの種類は決定されている。図10において、ステップS20では、書体コード（置換書体コードを含む）に基づいて選択したフォント名のフォントから文字コード（置換文字コードを含む）に対応するフォントデータを取得するため文字コードをチェックする。

【0035】一般的な日本語フォントの場合、文字コードは2バイトで表現されている。そこで、文字コードの上位半分と下位半分をそれぞれ検索キーとしてフォントデータ格納部12からフォントデータを取得する。文字コードに対応するフォントデータがない場合（ステップS21が肯定の場合）はステップS22に進み、フォントデータがない場合の代用コード、例えばブランクや「□」等に対応するコードを取得する。ステップS23では、文字コード（代用コードを含む）に対応するフォントデータを出力する。

【0036】次に、テーブル制御処理について詳述する。図11において、ステップS30では、テーブル制御データに基づいて処理内容つまり出力・更新・追加・削除のいずれであるかを判別する。処理内容はテーブル制御データのヘッダ領域から読み取る。ステップS31では対象テーブルを判別する。対象テーブルがフォント情報テーブル6ならばステップS32に進み、文字コード変換テーブル8ならばステップS32に進む。フォント情報テーブル6の制御では、処理内容に基づいて出力・更新・追加・削除のいずれの制御を行う。出力であればフォント情報テーブル6の内容を出力する。削除であればフォント情報テーブル6の内容を削除する。追加ならばテーブル制御データの内容をフォント情報テーブル

6に追加する。更新ならば、フォント情報テーブル6を一旦クリアにし、テーブル制御データの内容をフォント情報テーブル6に書き込む。

【0037】対象テーブルが文字コード変換テーブル8ならばステップS33に進む。文字コード変換テーブル8に関する制御も、フォント情報テーブル6の制御と同様であるので説明は省略する。なお、新たな文字またはマークを設定する場合には、文字コード変換テーブル8を変更して書体コード、文字コード列、置換文字コード、置換書体コード等を設定する必要があるが、フォント情報テーブル6は新規文字等の登録に際して変更する必要はない。

【0038】次に、文字コードのみの文書から書体コードを作成する前記書体コード作成部20の処理例を説明する。上記実施形態では、文字コードと書体コードとからなる文書ファイルがホストコンピュータ等、外部で作成したものが入力される場合を想定した。しかし、書体データが付加されていない文字コードのみの文書ファイルが入力される場合もあるので、この文字コードのみの文書ファイルから書体コードを作成したい場合も生じる。そこで、書体コード作成部20では、指定された文字の文字コードの書体コードをデフォルト以外のフォントを示すものに変換する処理を行う。

【0039】図12のフローチャートにおいて、ステップS100では、書体コード作成要求の有無を判別する。書体コード作成要求がない場合は、ステップS110に進み、デフォルトのフォントを採用して、文字データ獲得処理(ステップS120)に移行する。書体コード作成要求があった場合は、ステップS130に進み、フォント情報テーブルを参照し、使用できるフォントの種類をユーザに通知する。具体的には、当該文書処理装置の表示画面(図示せず)にフォント情報テーブル上のフォント名を表示する。この表示に対するユーザの応答つまり文字の指示とフォント名とが入力されれば(ステップS140が肯定)、ステップS150に進む。ステップS150では指示されたフォント名のフォント種類を示す書体コードを、指定された文字の書体コードとして出力する。該文字の文字コードと書体コードはそれぞれの登録領域に格納される。

【0040】書体コード作成処理を、図13のブロック図を参照してさらに説明する。図13において、前記作業メモリ2から文書ファイル19を読み込んで文字を表示部15に表示する。文字の表示はデフォルトの書体で表示される。ユーザは表示された文字毎に、書体を変更する文字を指定し、かつ表示されているフォント名を選択する。文字の指定や選択は、書体を変更する文字や表示されたフォント名にカーソルを合わせて予め機能が設定されたキーやマウス等の入力手段を操作する等、周知の入力方式によることができる。

【0041】こうして、選択されたフォントの種類つま

り書体に対応した書体コードをフォント情報テーブル16を参照して取得する。ユーザによって指定された文字には変更された書体の書体コードが割り付けられ、指定された以外の文字にはデフォルトの書体を示す書体コードが割り付けられ、それぞれの書体コードは記憶部17に格納され、文字コードは記憶部18に格納される。

【0042】なお、書体コード作成部20は文字コードのみからなる文書ファイルに関して書体コードを生成する場合に限らず、予め文字コードに対応づけられている書体コードを変更する場合にも同様に適用できる。

【0043】以上のように、本実施形態によれば、コード規格外の文字やマークをコードによって表現できる。文字等を表現する文字コードおよび書体コードのうち文字コードとしてはコード規格内のものを使用しているため、書体コードを解釈できない環境で文字等を出力しても、全く意味が通じない文書となることはない。

【0044】また、フォント情報テーブルによってコードチェック指示が検出された場合は、文字コード変換テーブルによって文字コード列が予定のコード列と一致しているかを判別して、単一の文字コードでは表現できないロゴマーク等を容易に出力させることができる。

【0045】なお、本発明は、プリンタ装置で文書を表示する場合だけでなく、ディスプレイ画面に文書を表示させる場合にも適用できる。例えば、ウインドウズ(Windows)環境に対応する形のフォントデータをフォントデータ格納部12に用意し、ラスターライズされたデータをウインドウのためのフレームバッファに展開するようにすれば、ウインドウの中に異字体やロゴマーク等を表示させることができる。

【0046】

【発明の効果】以上の説明から明らかなように、本発明によれば、文字コードに対応づけられる書体コードによって、コード規格に含まれない文字やパターンを前記文字コードに対応させて出力させることができる。

【図面の簡単な説明】

【図1】 本発明の一実施形態に係る文書処理装置の構成を示すブロック図である。

【図2】 本発明の一実施形態で処理される文書ファイルの一例を示す図である。

【図3】 文書ファイルの出力結果の一例を示す図である。

【図4】 テーブル制御データの一例を示す図である。

【図5】 フォント情報テーブルの一例を示す図である。

【図6】 文字コード変換テーブルの一例を示す図である。

【図7】 文字コード変換処理の一例を示すブロック図である。

【図8】 本発明の一実施例に係る文書処理のフローチャートである。

【図9】 フォント情報解析処理のフローチャートである。

【図10】 フォントデータ取得処理のフローチャートである。

【図11】 テーブル制御処理のフローチャートである。

【図12】 書体コード作成処理のフローチャートである。

【図13】 書体コード作成処理の一例を示すブロック図である。

【図14】 従来装置の一例を示すブロック図である。

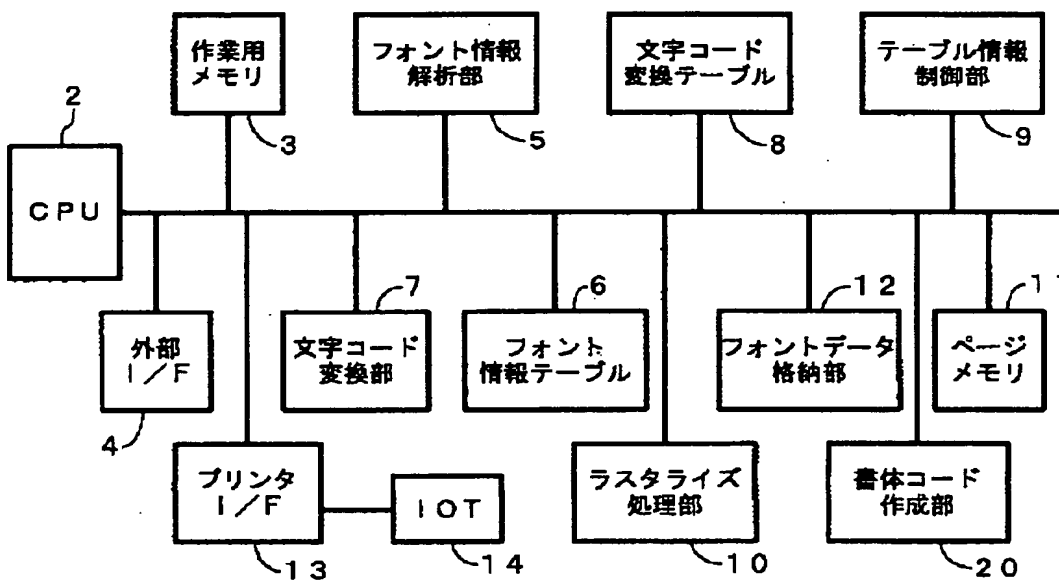
【図15】 従来装置の一例を示すブロック図である。

【図16】 従来装置の一例を示すブロック図である。

【符号の説明】

1…文書ファイル、 5…フォント情報解析部、 6…フォント情報テーブル、 7…文字コード変換部、 8…文字コード変換テーブル、 9…テーブル情報制御部、 12…フォントデータ格納部、 20…書体コード作成部

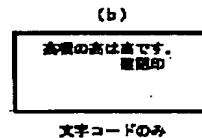
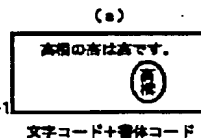
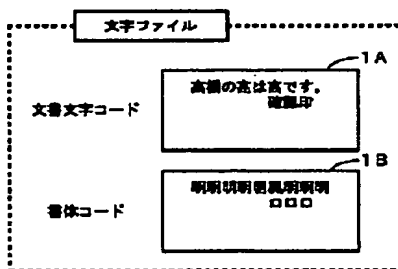
【図1】



【図2】

【図3】

【図4】



対象テーブル
制御内容
データ内容

| フォント情報 | |
|-------------------|--------------------|
| 更新 | |
| 明朝/Font/Min: | 明朝/Font/Min-1ji:明朝 |
| ゴシック/Font/Mark1: | 明朝:変換 |
| ゴシック/Font/Both1a: | |

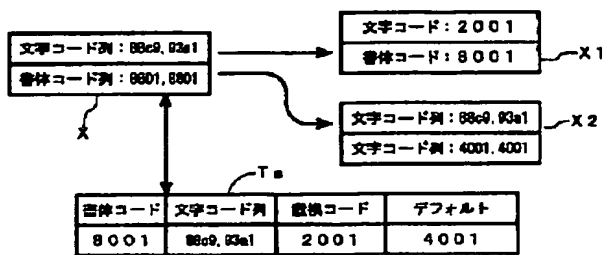
【図6】

【図5】

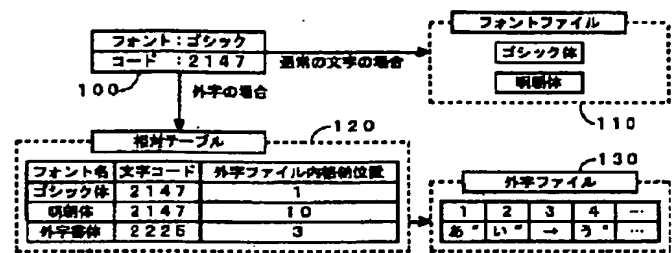
| No | フォント名 | 格納場所 | デフォルト | |
|----|-------|---------------------|-------|---|
| 1 | 明朝体 | /Font/Mincho.fnt | | |
| 2 | 明朝異字体 | /Font/Mincho-1.fnt | 1 | |
| 3 | ロゴマーク | /Font/LogoMark1.fnt | 1 | ○ |
| 4 | ゴシック体 | /Font/Gothic.fnt | | |

| 書体コード | 文字コード表 | 置換文字コード | 置換書体コード |
|-------|------------------|---------|---------|
| 3 | 88d1, 9443, 88f3 | 2421 | 4 |
| 3 | 88fa, 8200, 8abd | 2422 | 4 |

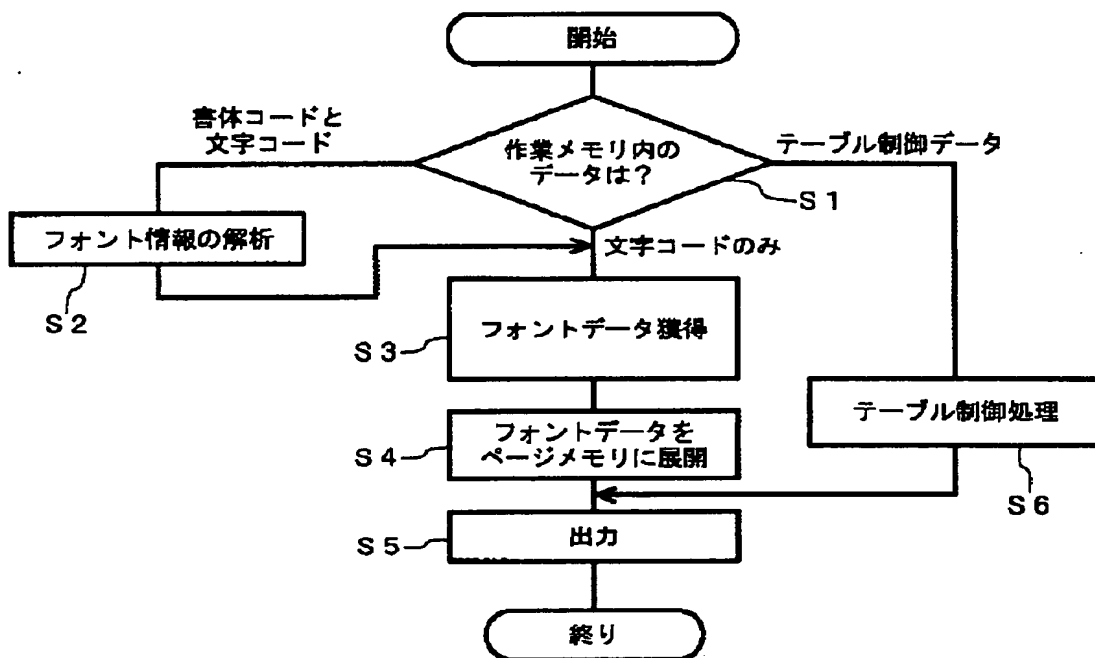
【図7】



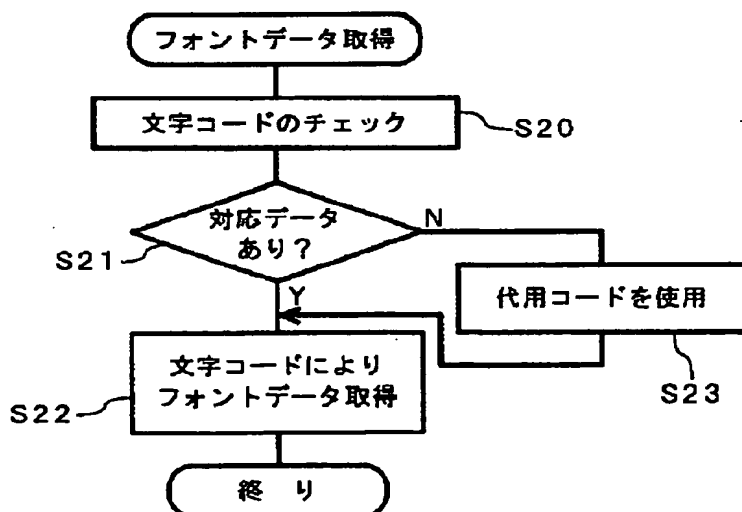
【図14】



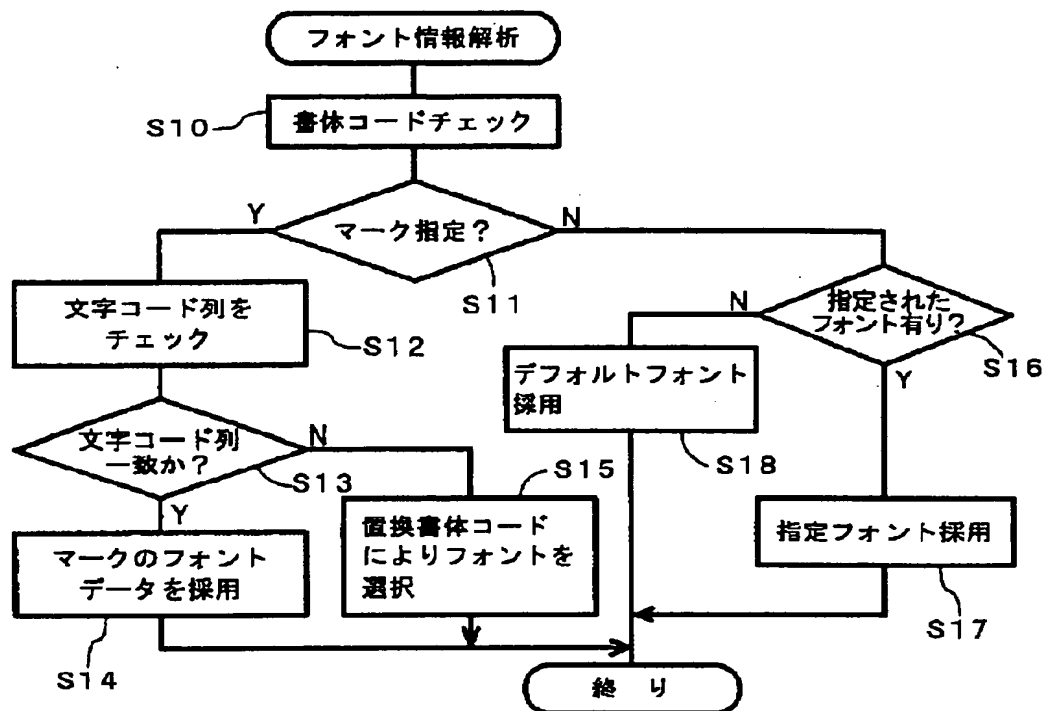
【図8】



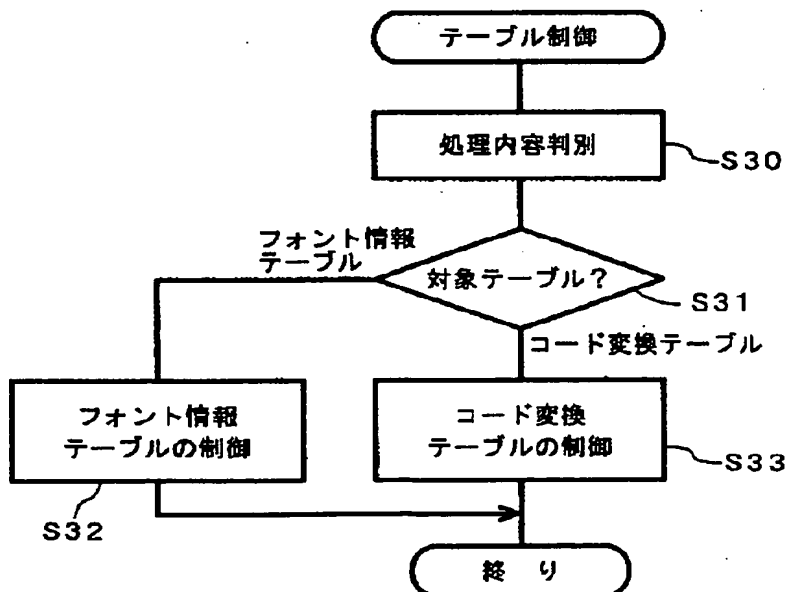
【図10】



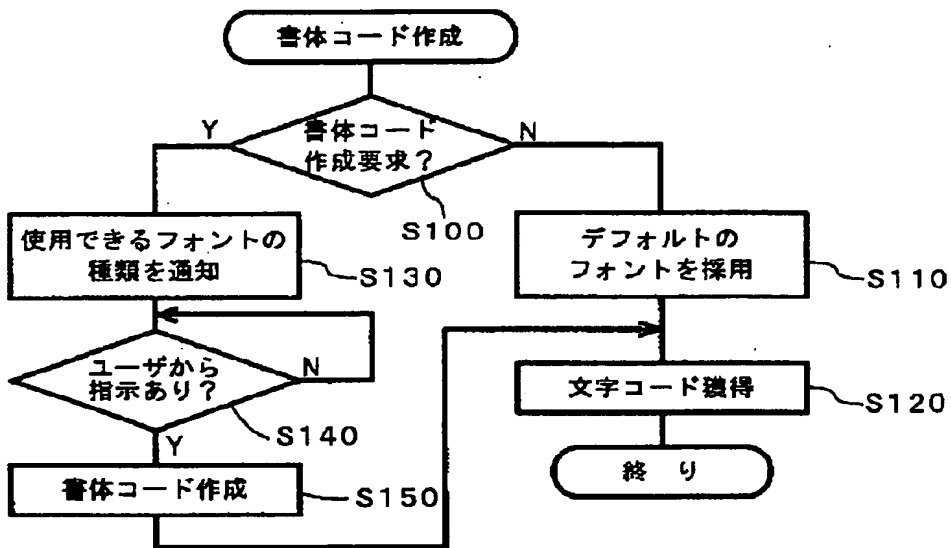
【図9】



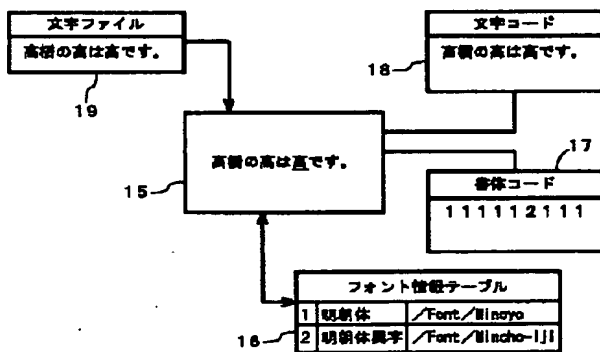
【図11】



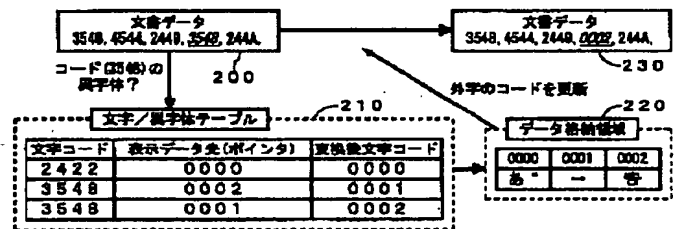
【図12】



【図13】



【図15】



【図16】

